

# 다중 태그 식별을 위한 개선된 질의 트리 충돌방지 알고리즘

임 인택<sup>†</sup>

## 요 약

본 논문에서는 RFID 시스템에서 다중 태그 식별을 위하여 질의 트리 기반의 QT\_rev 알고리즘을 제안한다. QT 알고리즘에서는 질의 문자열이 식별코드의 처음 비트들과 일치하는 태그는 전체의 식별코드로 응답한다. 반면, 본 논문에서 제안한 QT\_rev 알고리즘에서는 식별코드 중에서 질의 문자열을 제외한 나머지 비트들로만 응답한다. 또한 리더는 태그들의 응답 문자열 중에서 충돌이 발생한 비트 위치를 알 수 있으므로 충돌이 발생한 위치가 태그 식별코드의 마지막 비트이면 리더는 더 이상의 질의가 없이 두 개의 태그를 동시에 식별할 수 있다. 성능 분석의 결과, 본 논문에서 제안한 QT\_rev 알고리즘은 QT 알고리즘에 비하여 리더의 질의 횟수와 태그의 응답 비트 수가 월등히 적음을 알 수 있었다.

## Enhanced Query Tree Based Anti-Collision Algorithm for Multiple Tag Identification

In-Taek Lim<sup>†</sup>

## ABSTRACT

This paper proposes a QT\_rev algorithm for identifying multiple tags. The proposed QT\_rev algorithm is based on the query tree algorithm. In the QT algorithm, the tag will send all the bits of their identification codes when the query string matches the first bits of their identification codes. On the other hand, in the QT\_rev algorithm, the tag will send the remaining bits of their identification codes. After the reader receives all the responses of the tags, it knows which bit is collided. Therefore, if the collision occurs in the last bit, the reader can identify two tags simultaneously without further query. According to the simulation results, the QT\_rev algorithm outperforms the QT algorithm in terms of the number of queries and the number of response bits.

**Key words:** RFID(무선태그식별), Anti-collision algorithm(충돌방지 알고리즘), QT algorithm(질의트리 알고리즘)

## 1. 서 론

현재 RFID 시스템에서 사용되는 태그는 그 능력이 매우 제한적이어서 다른 태그들과 통신할 수 없고 단지 리더기와 통신할 수 있다. 리더는 무선채널을 통하여 각각의 태그들과 통신하는데, 모든 태그들은 리더가 보낸 신호를 동시에 듣게 된다. 리더로부터 요청 메시지를 받은 태그들은 동시에 리더로 자신의

식별코드를 전송하기 때문에 태그 충돌(Tag collision)이 발생한다[1,2]. 이 때 리더에서는 동시에 응답한 여러 개의 태그를 식별해야하는 문제가 발생하는데, 이를 해결하는 기술이 충돌방지(Anti-collision) 알고리즘이며, 이는 RFID 시스템에서 가장 핵심이 되는 기술이다[3,4].

RFID 시스템 성능의 두 가지 중요한 척도는 태그를 식별하기 위해 필요한 시간과 태그에 의해 소모되

※ 교신저자(Corresponding Author) : 임인택, 주소 : 부산광역시 남구 우암동 산55-1번지(608-738), 전화 : 051)640-3423, FAX : 051)640-3038, E-mail : itlim@pufs.ac.kr

접수일 : 2005년 7월 4일, 완료일 : 2005년 10월 4일

<sup>†</sup>중신회원, 부산외국어대학교 컴퓨터공학부

는 에너지이다. 리더에 의해 전송요구가 있을 때 태그가 보낸 데이터가 적을수록 식별 시간이 단축되고 태그에 의해 소모되는 에너지도 적다[5]. 따라서 충돌방지 알고리즘은 RFID 시스템의 태그 식별 속도 및 정확성을 결정하는 중요한 핵심기술이다[6,7].

충돌방지 알고리즘은 크게 확률적 알고리즘과 결정적 알고리즘으로 구분된다. 확률적 알고리즘은 13.56MHz 대역을 사용하는 EPC Class 1과 ISO 15693 표준에서 채택하고 있으며, 슬롯 ALOHA 방식을 기반으로 하고 있다. 확률적 알고리즘의 대표적인 방식으로는 FSA(Frame Slotted ALOHA) 알고리즘[6,10]이 있는데, 대부분의 확률적 알고리즘은 난수 발생기를 사용하여 태그가 응답하는 슬롯을 확률적으로 결정하므로 태그 식별의 완전성에서 제한점이 있다. 결정적 알고리즘은 915MHz 대역의 EPC Class 0에서 채택하고 있으며, 트리 검색 방식을 기반으로 하고 있다. 이는 다시 기억형 알고리즘과 무기억형 알고리즘으로 구분된다[11]. 기억형 알고리즘의 대표적인 방식은 Bit-by-Bit 이진트리 알고리즘 및 비트 증재 알고리즘이 있다. 이러한 알고리즘에서 태그의 응답은 태그에 대한 리더기의 질의와 태그의 현재 상태에 의하여 결정된다. 따라서 대부분의 기억형 결정적 알고리즘들은 태그 식별과정에서 태그들이 자신의 상태를 관리해야하므로 저가격·저전력·초소형화 문제를 해결하는 것이 가장 큰 문제점이 되고 있다. 무기억형 알고리즘의 대표적인 방식은 이진트리 검색 알고리즘과 QT(Query Tree) 알고리즘이 있는데, 태그의 응답은 리더기의 질의에 의해서만 결정된다. 따라서 무기억형 결정적 알고리즘은 경우 태그 구현이 간단하여 저가격·저전력·초소형화 문제를 해결할 수 있는 좋은 방법이지만, 낮은 성능을 해결해야 하는 문제점이 있다.

태그 식별을 위한 충돌방지 알고리즘 중에서 QT(Query-Tree) 알고리즘은 Auto-ID센터에서 제안한 무기억형(Memoryless) 알고리즘으로서 모든 태그들은 태그 식별과정에서 일어난 이전 질의의 이력을 기억할 필요가 없다[8]. QT 알고리즘인 경우, 리더가 질의한 질의 문자열인 프리픽스가 식별코드의 프리픽스와 일치하는 태그는 전체의 식별코드로 응답한다. 이로 인하여 태그가 전송해야하는 비트의 수가 많은 문제점이 있다. 본 논문에서는 이를 개선하여 응답하는 태그들은 식별코드 중에서 프리픽스를 제외한 나머지 비트들만으로 응답하는 QT\_rev(Revised

QT) 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 서론에 이어 2장에서는 본 논문에서 비교 분석하고자 하는 무기억형 알고리즘인 QT 알고리즘을 설명하고, 3장에서는 다중 태그 식별을 위하여 본 논문에서 제안하는 QT\_rev 알고리즘을 설명하고, 4장에서는 이들 알고리즘의 성능분석 결과를 기술하고, 마지막으로 결론을 맺는다.

## 2. QT 알고리즘

QT 알고리즘에서는 매 질의마다 식별코드의 프리픽스를 질의 문자열로 하여 전송한다. 수신한 질의 문자열이 식별코드의 프리픽스와 일치하는 태그는 자신의 전체 식별코드를 전송하고 리더로부터 다음의 질의를 기다린다. 반면 질의 문자열이 일치하지 않는 태그는 STAND-BY 상태로 천이하여 하나의 태그가 완전히 식별되어 다음 사이클이 시작될 때까지 리더의 질의에 응답하지 않는다[8].

전송한 질의 문자열에 대하여 둘 이상의 태그가 응답하면 충돌이 발생한다. 이 경우 리더는 이전의 질의 문자열에 비트 '0'과 '1'을 각각 연장하여 새로운 질의 문자열을 구성하여 다음 단계를 반복한다. 반면 리더가 충돌을 감지하지 않으면 하나의 태그를 완전히 식별한 경우이다. 이 경우, 리더는 질의 문자열을 갱신하고 다음 태그를 식별하기 위하여 위의 사이클을 반복한다.

QT 알고리즘의 동작은 다음과 같다. 먼저 태그의 식별코드 길이를  $k$  비트로 가정한다.  $A$ 를 최대 길이가  $k$  비트인 이진 문자열의 집합이라 하고,  $w$ 를 태그의 식별코드 문자열이라 하면, 집합  $A$ 와 문자열  $w$ 는 다음과 같이 각각 정의된다.

$$A = Y_{i=0}^k \{0,1\}^i \quad (1)$$

$$w \in \{0,1\}^k \quad (2)$$

한편 리더의 상태는  $A$ 의 문자열로 구성된 일련의 문자열인 큐  $Q$ 와  $A$ 의 원소들로 구성된 문자열의 집합인 메모리  $M$ 으로 구성된다. 여기서 큐  $Q$ 는 리더가 다음에 질의할 질의 문자열을 저장하는 용도로 사용되고, 메모리  $M$ 은 이미 식별된 태그의 식별코드를 저장하는 용도로 사용된다. 리더가 방송하는 질의는  $A$ 에 있는 문자열  $q(q \in A)$  정의하고, 태그의 응답은 태그의 식별코드 문자열인  $w$ 로 정의한다. 초기 상태

에서 리더의 큐  $Q$ 와 메모리  $M$ 은 비어있다. 먼저 리더의 알고리즘은 다음과 같다.

- 1) 큐를  $Q = \langle q_1, q_2, \dots, q_l \rangle$ 로 둔다.
- 2) 질의  $q_1$ 을 큐에서 꺼내서 방송한다.
- 3) 큐를  $Q = \langle q_2, \dots, q_l \rangle$ 로 갱신한다.
- 4) 태그로부터 수신한 응답에 대하여,
  - ① 응답이  $w$ 이면 태그  $w$ 를 식별한 것으로 표시하고,  $w$ 를 메모리  $M$ 에 삽입한다.
  - ② 만일 충돌이 발생했으면, 큐를  $Q = \langle q_2, \dots, q_l, q_1^0, q_1^1 \rangle$ 로 갱신한다.
- 4) 큐  $Q$ 가 빌 때까지 위의 과정을 반복한다.

태그의 알고리즘은 다음과 같다.

- 1) 태그의 식별코드  $w$ 를  $w = w_1 w_2 \dots w_k$ 라 한다.
- 2) 리더로부터 수신한 질의를  $q$ 라 하고,  $q$ 의 길이를  $|q|$ 라 한다.
- 3)  $q = \epsilon$  또는  $q = w_1 w_2 \dots w_{|q|}$ 이면, 태그는  $w$ 로 응답한다. 즉 질의  $q$ 가 자신의 식별코드 일부분과 일치하면 전체 식별코드를 리더로 보낸다.

그림 1은 QT 알고리즘에서 태그의 상태 천이도를 나타낸 것이다. STAND-BY 상태에 있는 태그가 리더로부터 질의 문자열을 수신하면 RECEIVING 상태로 천이한다. RECEIVING 상태에서 수신하는 질의 문자열이 식별코드의 프리픽스와 일치하지 않으면 STAND-BY 상태로 천이하여 다음 사이클의 질의를 기다린다. 반면 식별코드의 프리픽스가 질의 문자열과 일치하는 태그는 SENDING 상태로 천이하여 식별코드의 전체 비트를 리더로 전송한 후 STAND-BY 상태로 천이하여 다음 사이클의 질의를 기다린다.

표 1은 태그의 식별코드가 <0001, 0010, 1010, 1011>

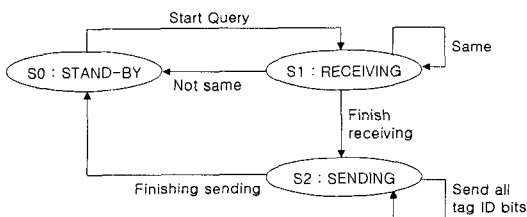


그림 1. QT 알고리즘의 태그 상태 천이도

표 1. QT 알고리즘의 동작

질의 문자열	태그 응답	응답한 태그	갱신한 큐
$\epsilon$	X0XX	all	<0,1>
0	00XX	0001,0010	<1,00,01>
1	101X	1010,1011	<00,01,10,11>
00	00XX	0001,0010	<01,10,11,000,001>
01	-		<10,11,000,001>
10	101X	1010,1011	<11,000,001,100,101>
11	-		<000,001,100,101>
000	0001	0001	<001,100,101>
001	0010	0010	<100,101>
100	-		<101>
101	101X	1010,1011	<1010,1011>
1010	1010	1010	<1011>
1011	1011	1011	< $\epsilon$ >

인 네 개의 태그를 식별하기 위한 QT 알고리즘의 동작을 나타낸 것이다. 표에서 나타낸 태그 응답에서 'X'는 충돌이 발생한 비트를 의미한다. 표에서 나타낸 바와 같이 네 개의 태그를 모두 식별하기 위하여 총 13번의 질의 사이클이 필요하다. 이 때 리더가 방송하는 질의 문자열의 총 비트 수는 30 비트이고, 모든 태그들이 전송하는 총 비트의 수는 72 비트이다.

### 3. QT\_rev 알고리즘

QT 알고리즘에서는 질의 문자열이 식별코드의 프리픽스와 일치하는 태그는 식별코드의 전체 비트를 전송한다. 따라서 식별코드의 길이가 길어지면 태그가 전송하는 데이터 량이 증가하고, 이는 태그의 전력 소모량을 증가시키는 결과를 초래한다. 또한 리더가 태그의 길이를 미리 알고 있고, 충돌이 발생한 비트 위치를 알 수 있다면 마지막 비트에서 충돌이 발생할 경우 두개의 태그를 동시에 식별할 수 있다. 그럼에도 불구하고 QT 알고리즘에서는 큐가 빌 때까지 모든 과정을 반복하므로 식별 시간이 증가하고, 이로 인하여 에너지 소모량이 증가하는 단점이 있다. 이러한 문제점을 해결하기 위하여 본 논문에서는 QT 알고리즘의 성능을 개선한 QT\_rev(Revised QT) 알고리즘을 제안한다.

본 논문에서 제안하는 QT\_rev 알고리즘인 경우,

리더는 태그의 응답 문자열 중에서 충돌이 발생한 비트 위치를 알 수 있다고 가정한다. QT\_rev 알고리즘에서는 QT 알고리즘에서와 같이 매 질의마다 식별코드의 프리픽스를 질의 문자열로 전송한다. 수신한 질의 문자열이 식별코드의 프리픽스와 일치하는 태그는 자신의 전체 식별코드를 전송하는 대신에 식별코드 중에서 질의 문자열에 해당하는 프리픽스를 제외한 나머지 문자열로 응답하고 리더로부터 다음의 질의를 기다린다. 반면 일치하지 않는 태그는 STAND-BY 상태로 천이하여 하나의 태그가 완전히 식별되어 다음 사이클이 시작될 때까지 리더의 질의에 응답하지 않는다.

전송한 질의 문자열에 대하여 둘 이상의 태그가 응답하면 충돌이 발생한다. 이 경우 리더는 이전의 질의 문자열에 충돌이 아닌 응답 문자열과 비트 '0', '1'을 각각 연장하여 새로운 질의 문자열을 구성하여 다음 단계를 반복한다. 또한 식별코드 중에서 마지막 비트에서 충돌이 발생한 경우, 리더는 마지막 비트만 다른 두 개의 태그가 식별영역 내에 존재한다는 것을 알 수 있으므로 두 개의 태그를 동시에 식별할 수 있다. 반면 리더가 충돌을 감지하지 않으면 하나의 태그를 완전히 식별한 경우이다. 이 경우, 리더는 질의 문자열을 갱신하고 다음 태그를 식별하기 위하여 위의 사이클을 반복한다.

QT\_rev 알고리즘의 동작은 다음과 같다. 리더의 질의 문자열을  $q(q \in A)$  정의하고,  $|q|$ 를 질의 문자열의 길이로 정의한다.  $w$ 를 태그의 식별코드 문자열이라 하고,  $r$ 을 태그의 응답 문자열이라 정의하면 태그의 응답 문자열은 태그의 식별코드 중에서 질의 문자열을 제외한 나머지 비트들로 구성된 문자열이므로 다음과 같이 정의된다.

$$r = r_{|q|+1} r_{|q|+2} \dots r_k \quad (3)$$

먼저 리더의 알고리즘은 다음과 같다.

- 1) 큐를  $Q = \langle q_1, q_2, \dots, q_l \rangle$  로 둔다.
- 2) 질의  $q_1$ 을 큐에서 꺼내서 방송한다.
- 3) 큐를  $Q = \langle q_2, \dots, q_l \rangle$ 로 갱신한다.
- 4) 태그로부터 수신한 응답에 대하여,
  - ① 태그로부터의 응답  $r$ 중에서 충돌이 아닌 문자열을  $z$ 라 하고, 문자열  $z$ 의 길이를  $|z|$ 라 한다.
  - ②  $|z| = k - |q|$ , 즉 충돌이 아니면 태그의 식별코

드  $q_1 r$ 을 식별한 것으로 표시하고 메모리에 삽입한다.

- ③  $|z| = k - |q| - 1$ , 즉 식별코드의 마지막 비트에서 충돌이면 태그의 식별코드  $q_1 r_{|q|+1} r_{|q|+2} \wedge r_{k-1} 0$ 과  $q_1 r_{|q|+1} r_{|q|+2} \wedge r_{k-1} 1$ 을 식별한 것으로 표시하고 메모리에 삽입한다.

- ④ 만일 충돌이 발생했으면, 큐를

$$Q = \langle q_2, \dots, q_l, q_1 z 0, q_1 z 1 \rangle$$

- 5) 큐  $Q$ 가 빌 때까지 위의 과정을 반복한다.

태그의 알고리즘은 다음과 같다.

- 1) 태그의 식별코드  $w$ 를  $w = w_1 w_2 \dots w_k$ 라 한다.
- 2) 리더로부터 수신한 질의를  $q$ 라 하고,  $q$ 의 길이를  $|q|$ 라 한다.
- 3)  $q = \varepsilon$  또는  $q = w_1 w_2 \dots w_{|q|}$ 이면, 태그는  $r = r_{|q|+1} r_{|q|+2} \wedge r_k$ 로 응답한다. 즉 질의  $q$ 가 자신의 식별코드 처음 비트들과 일치하면 식별코드 중에서 질의 문자열을 제외한 나머지 비트들을 리더로 보낸다.

그림 2는 QT\_rev 알고리즘에서 태그의 상태 천이도를 나타낸 것이다. STAND-BY 상태에 있는 태그가 리더로부터 질의 문자열을 수신하고, 수신한 질의 문자열이 식별코드의 프리픽스와 일치하는 태그는 SENDING 상태로 천이한다. SENDING 상태에 있는 태그는 QT 알고리즘과는 달리 자신의 식별코드 중에서 질의 문자열을 제외한 나머지 비트들만으로 응답한다.

표 2는 태그의 식별코드가 <0001, 0010, 1010, 1011>인 네 개의 태그를 식별하기 위한 QT\_rev 알고리즘의 동작을 나타낸 것이다. 표에서 나타낸 바와 같이 네 개의 태그를 모두 식별하기 위하여 총 5번의 질의 사이클이 필요하다. 이 때 리더가 방송하는 질

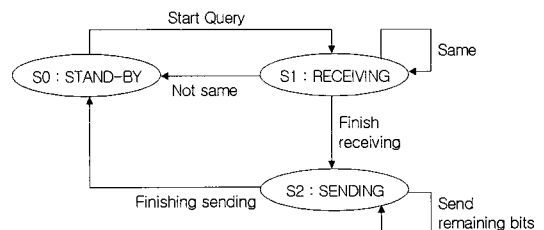


그림 2. QT\_rev 알고리즘의 태그 상태 천이도

표 2. QT\_rev 알고리즘의 동작

질의 문자열	태그 응답	응답한 태그	갱신된 큐
$\epsilon$	X0XX	all	<0,1>
0	0XX	0001,0010	<1,000,001>
1	01X	1010,1011	<000,001>
000	1	0001	<001>
001	0	0010	< $\epsilon$ >

의 문자열의 총 비트 수는 8 비트이고, 모든 태그들이 전송하는 총 비트의 수는 30 비트이다.

표 3은 4개의 태그를 식별하기 위하여 표 1과 표 2에서 각각 나타낸 QT 알고리즘과 QT\_rev 알고리즘의 동작 과정에서 얻어진 질의 횟수, 질의 비트의 수, 및 응답 비트의 수를 요약한 것이다. 표에서 나타낸 바와 같이 본 논문에서 제안한 QT\_rev 알고리즘이 QT 알고리즘에 비하여 모든 성능 요소에서 우수함을 알 수 있다.

QT\_rev 알고리즘에서는 QT 알고리즘과는 달리 태그들의 응답 문자열 중에서 충돌이 발생한 비트 위치를 리더가 알 수 있다. 이로 인하여 태그들로부터 수신한 응답 문자열에서 충돌이 발생했고, 충돌이 발생한 위치가 태그 식별코드의 마지막 비트이면 리더는 두 개의 태그를 동시에 식별한다. 따라서 리더의 식별영역 내에 있는 모든 태그들을 식별하는 시간을 줄일 수 있다. 또한 충돌이 발생한 위치가 태그 식별코드의 마지막 비트가 아니면 이전의 질의 문자열에 충돌이 아닌 응답 문자열과 비트 '0', '1'을 각각 추가하여 다음에 질의할 질의 문자열로 한다. 따라서 리더는 태그 식별코드의 일부분을 미리 알 수 있으므로 질의 횟수를 줄일 수 있는 장점이 있다.

표 3. 예를 통한 두 알고리즘의 비교

알고리즘	질의 횟수	질의 비트 수	응답 비트 수
QT	13	30	72
QT_rev	5	8	30

### 4. 성능 평가

본 논문에서는 시뮬레이션을 통하여 QT 알고리즘과 QT\_rev 알고리즘의 성능을 비교하였다. 시뮬레이션을 위하여 태그의 식별코드 길이는 64비트로

가정하였다. 태그의 식별코드가 무작위인 경우와 순차적인 경우 각각에 대하여 식별영역 내의 모든 태그를 식별하기 위하여 태그의 수에 따른 리더의 질의 횟수 및 모든 태그가 보낸 비트 수를 성능평가 매개변수로 하였다. 식별코드가 순차적인 경우는 식별코드의 최하위 비트부터 순차적으로 증가하는 것으로 가정하였다.

그림 3과 4는 태그의 식별코드를 무작위로 한 경우와 순차적으로 한 경우, 모든 태그를 식별하기 위한 총 질의 횟수를 각각 나타낸 것이다. 태그의 식별코드가 무작위인 경우, 500개의 태그를 식별하기 위하여 QT 알고리즘은 1,453번의 질의를 필요로 하고, QT\_rev 알고리즘은 999번의 질의를 필요로 한다. 따라서 본 논문에서 제안한 QT\_rev 알고리즘이 QT 알고리즘에 비하여 약 1.5배의 성능 향상을 얻을 수 있었다. 이는 QT\_rev 알고리즘인 경우, 충돌이 발생한 비트의 위치를 리더가 알 수 있기 때문이다. 또한 태그의 식별코드가 순차적인 경우, 500개의 태그를

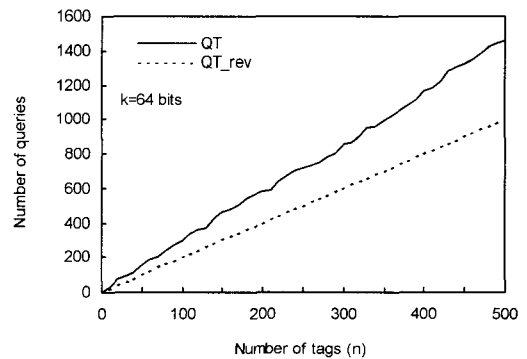


그림 3. 총 질의 횟수(무작위 식별코드)

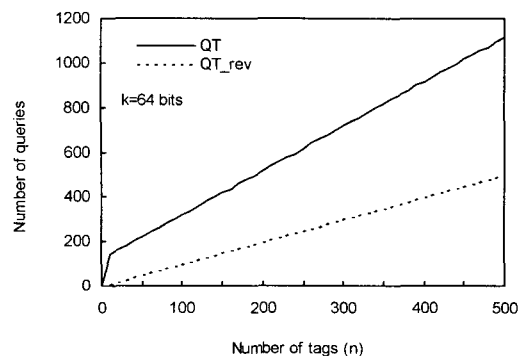


그림 4. 총 질의 횟수(순차적 식별코드)

식별하기 위하여 QT 알고리즘 1,113번의 질의를 필요로 하고, QT\_rev 알고리즘은 499번의 질의를 필요로 하므로 약 2.2배의 성능 향상을 얻을 수 있었다.

그림 5와 6은 태그의 식별코드를 무작위로 한 경우와 순차적으로 한 경우, 모든 태그를 식별할 때까지 태그들이 응답한 총 비트의 수를 각각 나타낸 것이다. 태그의 식별코드가 무작위이든 순차적이든 상관없이 본 논문에서 제안한 QT\_rev 알고리즘이 QT 알고리즘에 비하여 태그가 응답하는 비트의 수가 적다. 이는 QT 알고리즘인 경우에는 질의 문자열과 일치하는 식별코드 프리픽스를 갖는 태그는 자신의 전체 식별코드로 응답하는 반면, 본 논문에서 제안한 QT\_rev 알고리즘인 경우에는 식별코드 중에서 질의 문자열을 제외한 나머지 비트들로만 응답하기 때문이다. 태그의 식별코드가 순차적일 때 태그가 전송하는 총 비트의 수는 QT 알고리즘인 경우 본 논문에서 제안한 QT\_rev 알고리즘에 비하여 약 50 배 더 많은 비트들을 전송하여야 한다. 이는 리더는 충돌이 발생한 비트의 위치를 알 수 있어서 마지막 비트에서 충

돌이 발생한 경우 리더는 두 개의 태그를 동시에 식별하기 때문이다. 따라서 모든 태그를 식별하기 위하여 태그가 보낸 총 비트의 수가 본 논문에서 제안한 QT\_rev 알고리즘이 QT 알고리즘에 비하여 약 1/50 이므로 리더의 식별영역 내의 모든 태그를 식별하기 위한 시간이 단축되고, 태그의 전력 소모량도 적음을 알 수 있다.

Auto-ID 센터의 클래스 0 RFID 시스템에서는 세 가지 형태의 64비트와 한 가지 형태의 96비트, 및 세 가지 형태의 256비트를 갖는 EPC(Electronic Product Code)를 정의하고 있다[9]. 일반적으로 식별코드의 길이가 길면 길수록 태그들이 응답해야 하는 비트의 수는 많아질 것으로 예상된다. 따라서 본 논문에서는 식별코드의 길이가 AutoID 센터에서 정의한 64, 96, 및 256비트일 때 태그 식별 알고리즘의 성능을 분석하였다. 표 4는 식별영역 내의 태그 수가 100개일 때, 식별코드의 길이에 따른 질의 횟수를 나타낸 것이다. 표에서 나타낸 바와 같이 식별코드가 무작위인 경우에는 식별코드 길이에 상관없이 두 알고리즘 모두 일정한 질의 횟수를 나타낸다. 또한 식별코드가 순차적인 경우 본 논문에서 제안한 QT\_rev 알고리즘은 길이에 관계없이 질의 횟수는 일정하다. 반면, QT 알고리즘의 경우, 길이가 64비트에서 256비트로 증가하면 약 2.2배의 질의 횟수를 더 필요로 한다. 표 5는 식별코드의 길이에 따른 태그들의 총 응답 비트 수를 나타낸 것이다. 표에서 나타낸 바와

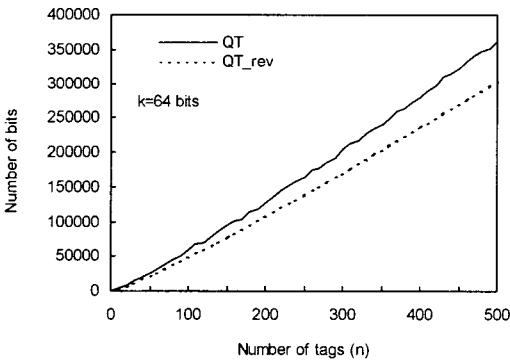


그림 5. 총 응답 비트의 수(무작위 식별코드)

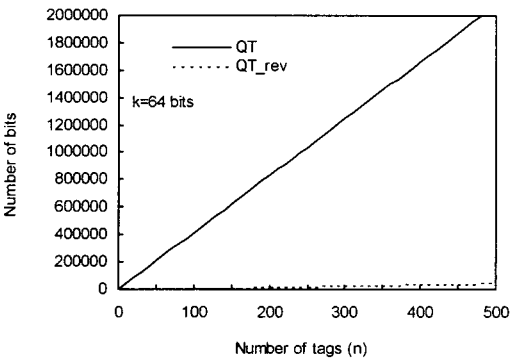


그림 6. 총 응답 비트의 수(순차적 식별코드)

표 4. 식별코드 길이 별 총 질의 횟수

길이(비트)	무작위 식별코드		순차적 식별코드	
	QT	QT_rev	QT	QT_rev
64	299	199	319	99
96	299	199	383	99
256	299	199	703	99

표 5. 식별코드 길이 별 총 응답 비트 수

길이(비트)	무작위 식별코드		순차적 식별코드	
	QT	QT_rev	QT	QT_rev
64	57,929	47,949	416,000	8,464
96	85,906	73,286	931,200	11,664
256	229,193	203,072	6,579,200	27,664

같이 식별코드가 순차적인 경우, 길이가 64비트에서 256비트로 증가하면 QT 알고리즘의 응답 비트 수는 약 15.8배로 증가하는 반면, 본 논문에서 제안한 QT\_rev 알고리즘은 약 3.2배만 증가한다. 따라서 식별영역 내에 있는 태그들의 식별코드가 순차적인 RFID 응용에서는 본 논문에서 제안한 QT\_rev 알고리즘이 QT 알고리즘에 비하여 월등히 우수한 성능을 나타낼 수 있음을 알 수 있다.

## 5. 결 론

본 논문에서는 무기억형 충돌방지 알고리즘인 QT 알고리즘을 개선한 QT\_rev 알고리즘을 제안하고, 이에 대한 성능을 분석하였다. QT 알고리즘인 경우, 리더가 방송한 질의 문자열이 식별코드의 프리픽스와 일치하는 태그는 자신의 식별코드 전체 비트를 질의에 대한 응답으로 전송한다. 따라서 식별코드의 길이가 길어지면 태그가 전송하는 데이터 량이 증가하고, 이에 따라 태그의 전력 소모량을 증가하는 문제점이 있다. 또한 식별코드의 마지막 비트가 충돌일 경우 두 개의 태그를 동시에 식별할 수 있음에도 불구하고, 리더는 충돌이 발생한 비트 위치를 알 수 없으므로 질의 사이클을 반복해야하는 문제점이 있다.

반면 본 논문에서 제안한 QT\_rev 알고리즘에서는 질의 문자열이 식별코드의 프리픽스와 일치하는 태그는 식별코드 중에서 질의 문자열을 제외한 나머지 비트들만으로 응답한다. 이로 인하여 태그들이 전송하는 비트의 수가 QT 알고리즘에 비하여 월등히 적음을 시뮬레이션을 통하여 알 수 있었다. 또한 리더는 태그들의 응답 문자열 중에서 충돌이 발생한 비트 위치를 알 수 있다. 태그들로부터 수신한 응답 문자열에서 충돌이 발생했고, 충돌이 발생한 위치가 태그 식별코드의 마지막 비트이면 리더는 두 개의 태그를 동시에 식별할 수 있으므로 QT\_rev 알고리즘에서는 리더의 질의 횟수를 줄일 수 있다. 성능 분석의 결과, 본 논문에서 제안한 QT\_rev 알고리즘은 QT 알고리즘에 비하여 리더의 질의 횟수와 태그의 응답 비트 수가 월등히 적음을 알 수 있었다. 따라서 본 논문에서 제안한 QT\_rev 알고리즘은 QT 알고리즘에 비하여 빠른 태그 식별이 가능하며 전력 소모량을 줄일 수 있는 장점이 있다.

## 참 고 문 헌

- [1] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, Carl Hanser Verlag GmbH & Co., 2002.
- [2] S. E. Sarma, S. A. Weis, and D. W. Engels, "RFID Systems and Security and Privacy Implications," *CHES2002, LNCS 2523*, pp. 454-469, Springer-Verlag, 2003.
- [3] P. Hernandez, J. D. Sandoval, F. Puente, and F. Perez, "Mathematical Model for a Multiread Anticollision Protocol," *Proc. 2001 IEEE Pacific Rim Conf. on Communications, Computers, and Signal Processing*, pp. 647-650, Aug. 2001.
- [4] 이근호, "무선식별(RFID) 기술," TTA저널, 제 89호, pp. 124-129, 2003. 10.
- [5] I. Chlamtac, C. Petrioli, and J. Redi "Energy-Conserving Access Protocols for Identification Networks," *IEEE/ACM Trans. Networking*, Vol. 7, No. 1, pp. 51-59, Feb. 1999.
- [6] H. Vogt, "Efficient Object Identification with Passive RFID Tags," *First International Conf. on Pervasive Computing, LNCS 2414*, pp. 99-113, Springer-Verlag, 2002.
- [7] M. Jacomet, A. Ehrsam, and U. Gehrigm "Contactless Identification Device with Anti-collision Algorithm," *Proc. IEEE CSCC'99*, Athenes Ltaly, July 1999.
- [8] C. Law, L. Lee, and K. Y. Siu, "Efficient Memoryless Protocol for Tag Identification," Auto-ID Center, *MIT-AUTOID-TR-003*, Oct. 2000.
- [9] Auto-ID Center, "860MHz-930MHz Class 0 Radio Frequency Identification Tag Protocol Specification Candidate Recommendation, Version 1.0.0," June 2003.
- [10] H. Vogt, "Multiple Object Identification with Passive RFID Tags," *Proc. IEEE International Conf. on Systems, Man and Cybernetics*, Vol. 3, pp. 658-663, Oct. 2002.

- [11] 김희철, 권성호, “대규모 RFID 시스템을 위한 충돌방지 알고리즘의 성능분석,” *Telecommunication Review*, 제15권 2호, pp. 274-289, 2005. 4.



임 인 택

1984년 울산대학교 전자계산학과(공학사)

1986년 서울대학교 대학원 계산통계학과 (이학석사)

1998년 울산대학교 대학원 컴퓨터공학과(공학박사)

1986년 ~ 1993년 : 삼성전자(주)

특수연구소 선임연구원

1993년 ~ 1998년 : 동부산대학 전자계산과 조교수

1998년 3월 ~ 현재 : 부산외국어대학교 컴퓨터공학부 부교수

관심분야 : MAC 프로토콜, 센서네트워크, RFID 응용